A&QT-R 2002 (THETA 13)
2002 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics
May 23-25, 2002, Cluj-Napoca, Romania

IMAGE PROCESSING APPLICATION USING THE JAVA PLATFORM AND THE JAVA ADVANCED IMAGING API

Authors: Mircea-Florin Vaida, Florin-Ovidiu Hurducas

Technical University of Cluj-Napoca, Gh. Baritiu str. 25-27, Romania
E-mail: Mircea.Vaida@com.utcluj.ro , hfovi@yahoo.com

ABSTRACT: Building applications using the Java Platform have the major advantage of simplicity and rapid development. The Java Platform has besides standard APIs (Application Programming Interface), a set of extended API like the Java Media APIs that imbue the Java 2 Standard Edition capabilities. Java Advanced Imaging is a component of the Java Media API group and provides imaging functionalities for Java applications. Another major advantage of Java Platform is the platform independence. Thinking to these advantages, we have developed an image processing application that has a user-friendly interface. The processing available in application is about acquiring, displaying, and creating (recording) images.

Keywords: Java 2 Standard Edition, Java Advanced Imaging, Java Media API, Image Processing System, image enhancement, image filtering

1. INTRODUCTION

The Java Advanced Imaging (JAI) API further extends the Java platform (including the Java 2D API) by allowing sophisticated, high-performance image processing to be incorporated into Java applets and applications. JAI is a set of classes providing imaging functionality beyond that of Java 2D and the Java Foundation Classes, though it is compatible with those APIs [1].

JAI implements a set of core image processing capabilities including image tiling, regions of interest, and deferred execution. JAI also offers a set of core image processing operators including many common points, area, and frequency-domain operators.

JAI is intended to meet the needs of all imaging applications. The API is highly extensible, allowing new image processing operations to be added in such a way as to appear to be a native part of it.

The JAI API specifies a core set of image processing operators. These operators provide a common ground for applications programmers, since they can then make assumptions about what operators are guaranteed to be present on all platforms.

The JAI API builds on the foundation of the Java 2D API to allow more powerful and general imaging applications. The JAI API adds the following concepts:

- Multi-tiled images

- Deferred execution
- Networked images
- Image property management
- Image operators with multiple sources
- Three-dimensional image data

The combination of tiling and deferred execution allows for considerable run-time optimization while maintaining a simple imaging model for programmers. New operators may be added and the new operators may participate as first-class objects in the deferred execution model.

The general categories of image processing operators supported include:

- Point Operators
- Area Operators
- Geometric Operators
- Color Quantization Operators
- File Operators
- Frequency Operators
- Statistical Operators
- Edge Extraction Operators
- Miscellaneous Operators

2. THE IMAGE PROCESSING APPLICATION

The application's goal is to perform a set of image processing operations on images of various formats. The application will be integrated in a dedicated radiology medical image processing application.

The application was built using JDK 1.4 development environment and JAI API 1.1. Another standard technology used in application is *JavaHelp*, which is an optional API shipped separately from JDK, that is specialized for developing and displaying the help content [2].

The application executes basic operations, which is part of any image processing application like acquiring, displaying, and creating images. The operations are performed on images taken from image files. These operations are controlled by a multi document user-friendly interface and are completed by a variety of functionalities for graphic editing, like undo/redo operations, working with selections, copying image segments, etc.

Extending the application capabilities is provided by the custom defined filters saving option and the possibility to use this filters when needed.

The application has implemented options for internationalization, like multilingual support and contextual help.

The image file formats supported by application for reading are: BMP (*Microsoft Windows bitmap*), FPX (*FlashPix*), GIF (*Graphics Interchange Format*), JPEG (*Joint Photographic Experts Group*), PNG (*Portable Network Graphics*), PNM (*Portable aNy Map*, including PBM, PGM, and PPM), TIFF (*Tag Image File Format*). The application support saving in the following image file formats: BMP, JPEG, PNG, PNM, and TIFF.

The users have the possibility to open image files coded in a specific format, and after processing they can save the images on disk in any supported format with the desired parameters, like compressing factor, image quality, etc.

After starting the application, the main window will be shown (see **Figure 1**). This window has the following elements:

- The menu bar – contains the menus which the user can access it for executing various editing actions and image processing. The inner menus are grouped by functionalities. There are menus specialized in working with files (opening, closing, saving, printing, etc.), menus specialized in graphic editing of images (section of image segments, undo/redo operations, etc.), menus specialized in image processing (various processing operations, histogram visualization, filtering, etc.), and the help menu which provides information about application and a user guide.
- The tool bar – these buttons have different functionalities that can be called from menus, but the positioning makes them easy to access.
- The desktop – this area is the working surface and holds the inner windows, which represents the image documents that the user are working on and can interact.
- The status bar – this area displays information concerning the program status or information used in graphic editing.

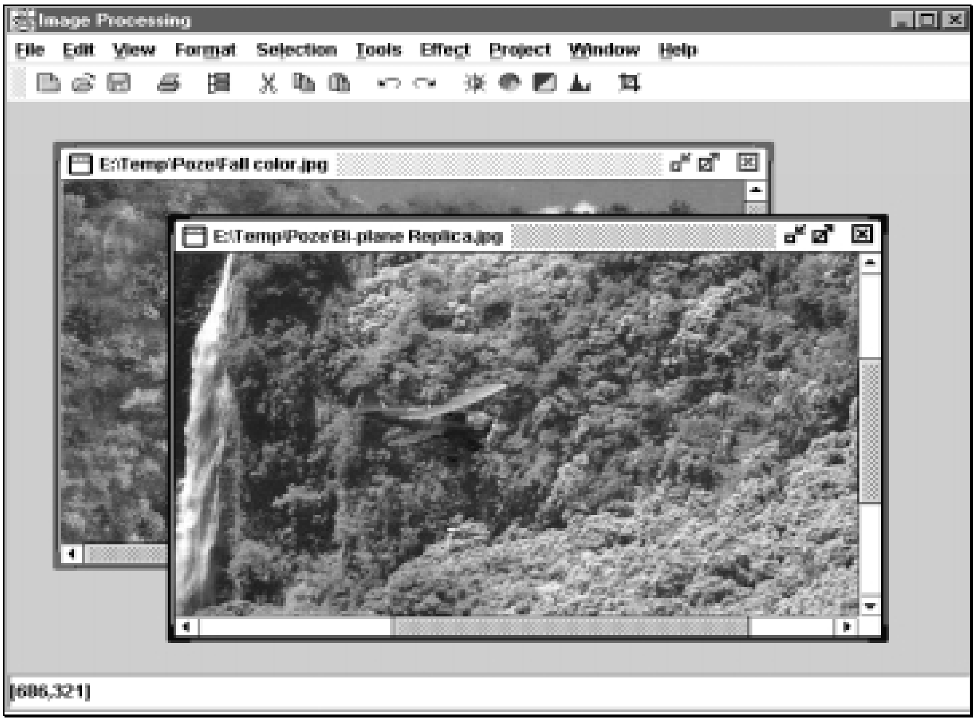
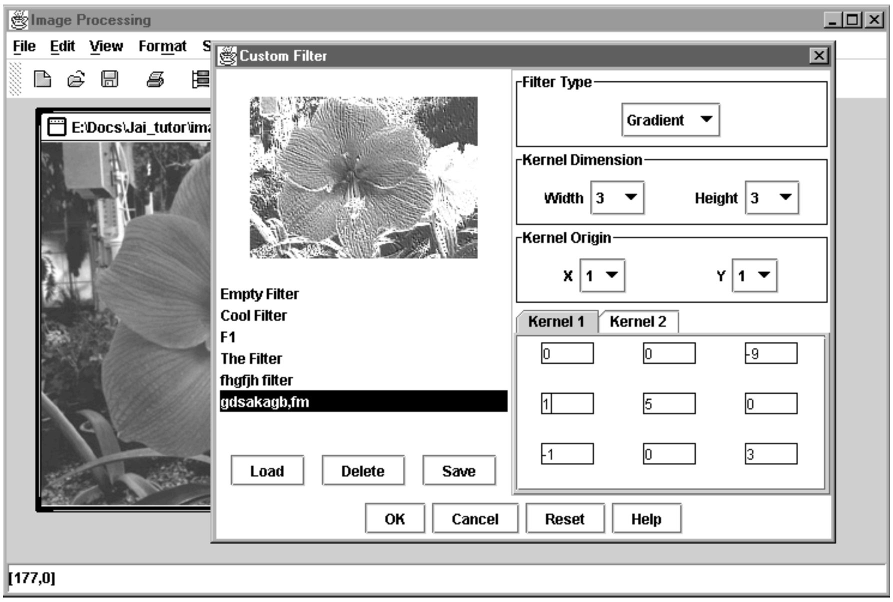


Figure 1. *The image processing application's main window*

The most important image processing operations that can be performed on open files are: cropping, rotating, transposing (vertically, horizontally, on diagonal, by 90, by 180, by 270), brightening adjust, color components modification, logarithmic enhancement, inverting, histogram analysis, processing using the histogram, image scaling, convolution filtering, gradient filtering, median filtering, threshold, etc.

Besides existing filters, the user can create new filters that can be applied on the current document or can be saved for future using (see **Figure 2.**).



Empty Filter
Cool Filter
F1
The Filter
fhgfjh filter
gdsakagb,fm

Filter Type: Gradient

Kernel Dimension: Width 3, Height 3

Kernel Origin: X 1, Y 1

0	0	-9
1	5	0
-1	0	3

[177,0]

Figure 2. *Creating user defined filters*

3. APPLICATION STRUCTURE

The application is structured on different blocks (see **Figure 3.**):

- The user interface – is the module that interprets the user commands to the application to execute an action (ex: image processing, loading a file, graphic editing, etc.).
- The control logic – is a block that manages different resources in program and provides functioning rules.
- The displaying component – renders the image and other graphical elements (ex. selections) and has functions for graphical editing (ex. undo/redo, resize monitoring of a image document, etc.).
- Processing operations – this block contain all the image processing operations, like point operators, area operators, geometric operators, file operators, frequency operators, statistical operators, etc

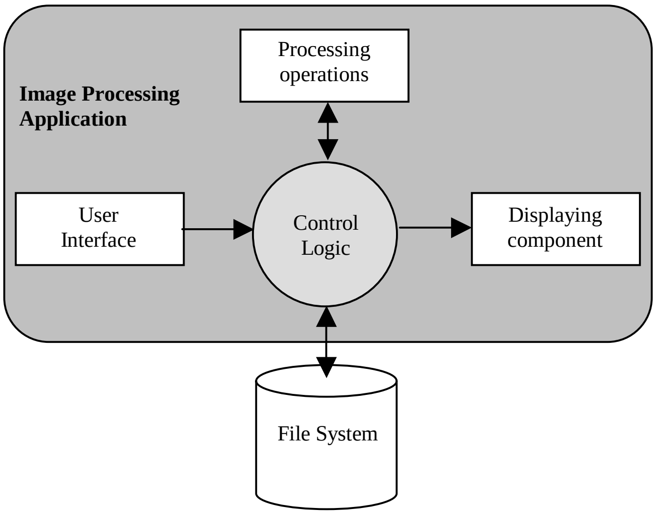


Figure 3. *Block schematics of the application*

The *User Interface* block captures the user input actions and the *Control Logic* block performs the right operations using the *Processing Operations* block (ex. loading a image document from the file system, executing a image processing operation, etc.), then display the results using the *Displaying Component*.

The model for image processing that take place in the application is shown in **Figure 4. [3]**

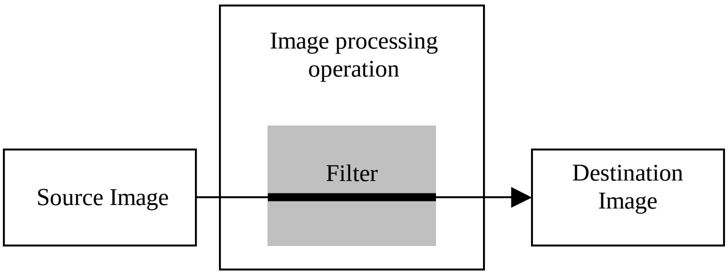


Figure 4. *Image processing model*

The processing chain that performs the image processing on an image file using a specified operator is shown in **Figure 5.** The data flows trough chain only when the displaying component requests this, and until that no image processing takes place.

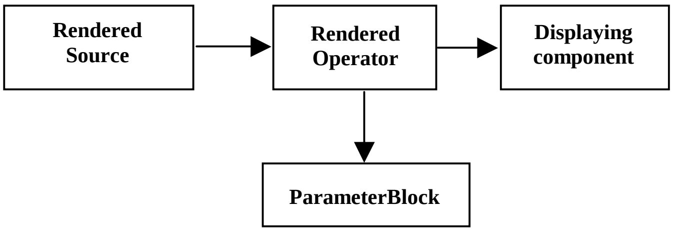


Figure 5. *Processing chain*

4. ADVANTAGES AND DISADVANTAGES OF THE APPLICATION

The *advantages* of the image processing application are:

- The application runs on any system with Java Platform and JAI API installed. The Java Platform provide for full portability of the application.
- The saved images are in BMP, JPEG, PNG, PNM, and TIFF.
- The image can be saved from a loaded format (including FPX and GIF) to any supported format (format translation).
- Hardware support on Windows and Solaris platforms for image processing operators. This increases the speed that some operators work.
- Simplicity in developing and extending the application
- Multi documents support

- User defined filters
- Working with selections

The *disadvantages* of the Image Processing application are:

- Medium to high hardware requirements because the application is making a lot of processing.
- On some platforms that not support JAI with hardware accelerated functions the application will work slowly
- The FPX and GIF format is not supported for saving

5. CONCLUSIONS

The application requires a computer with Java runtime and a capable processor. If the operating system is Windows or Solaris the image processing can be assisted by hardware accelerated operators sustained by MMX and VIS instructions.

The structure of the application allows extending the processing functionalities and can be used as an application framework for any image processing application. Also could be implemented a module for image analysis.

The formats for image files supported by application are FPX, GIF, BMP, JPEG, PNG, PNM, and TIFF. Others formats can be added if the new Image I/O technology is used, the rest of the application can remain the same because are compatible with that technology.

REFERENCES

[1] “*Programming in Java Advanced Imaging*”, Sun Microsystems, http://java.sun.com/products/java-media/jai, November 1999

[2], M.-F. Vaida, C. Porumb, R. Fotea, F.-O. Hurducas, L. Lazar, Java 2 Enterprise Edition (J2EE). Multimedia Applications, Edit. Albastra, 2002, Cluj-Napoca

[3], “*The JAI Tutorial*”, Sun Microsystems, http://java.sun.com/products-/java-media/jai, 1999